

Taller Windows 10 IoT Core + IoT Hub

Microsoft Developer eXperience

Contenido

Introducción	2
Paso 1: Montar el circuito del sensor.....	2
Paso 2: Creación de recursos en Azure	4
Creación del IoT Hub	5
Creación de Event Hubs	6
Creación de Stream Analytics.....	9
Paso 3: Registro del dispositivo.....	16
Paso 4: Código de Windows 10 IoT Core.....	18
Despliegue del código en la Raspberry Pi 2	19
Paso 5: monitorización.....	20
Paso 6a: Power BI.....	22
Paso 6b: Website.....	24
Paso 7: Retos	26
Anexo: activación Azure Pass	26

Introducción

Durante este taller vamos a utilizar **Windows 10 IoT Core** en una **Raspberry Pi 2** para desplegar una aplicación universal que leerá la información de un sensor de temperatura y humedad DHT22. Los datos del sensor se recogen cada dos segundos y se envían a Azure al servicio **IoT Hub** que permite realizar la ingestión masiva de eventos de forma segura.

También utilizaremos el servicio **Stream Analytics** para extraer información en tiempo real sobre los datos que enviamos al IoT Hub. Crearemos alertas y agregaciones sobre esos datos.

El ejercicio llega hasta la visualización de los datos mediante dos sistemas diferentes:

- El primero, un cuadro de mandos de **PowerBI** que nos permite crear una gráfica de forma muy sencilla. Para poder conectarlo necesitaremos crear un stream especial de Stream Analytics
- En el segundo caso, vamos a crear una **Web App** en Azure para desplegar un sitio web que lee del IoT Hub directamente y dibuja la gráfica en cliente con la librería **d3js**.

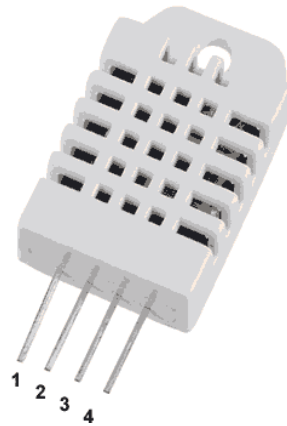
Antes de comenzar debemos comprobar que tenemos en nuestro ordenador:

- Windows 7 o posterior (preferiblemente Windows 10, actualízate!).
- Visual Studio 2015 Update 1 (la edición [Community](#) es suficiente), hay que instalar las herramientas de UWP.
- Windows IoT Core Project [Templates](#).
- El modo desarrollador habilitado (para habilitarlo hemos de ir a Configuration en nuestro ordenador → Update and Security → Developer mode).
- [Windows IoT Core Dashboard](#).
- [SDK de Azure](#).
- IoT Hub [Device Explorer](#).

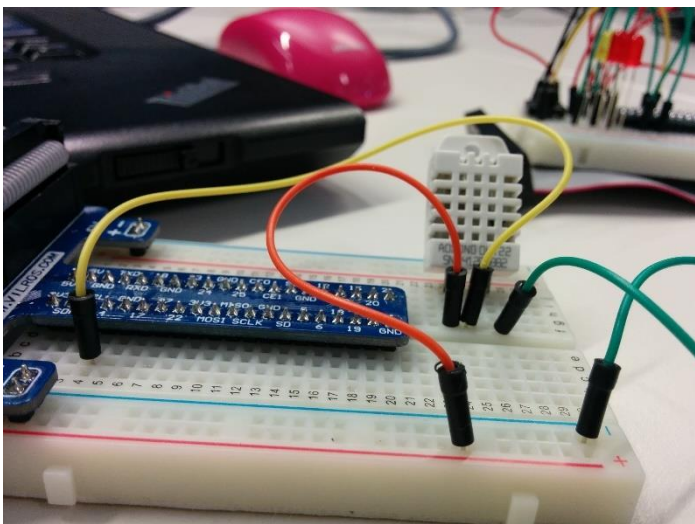
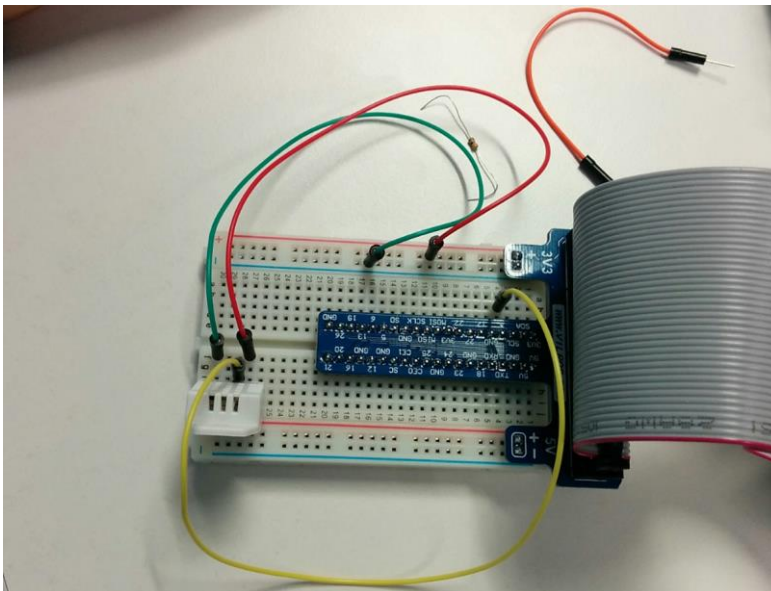
Paso 1: Montar el circuito del sensor

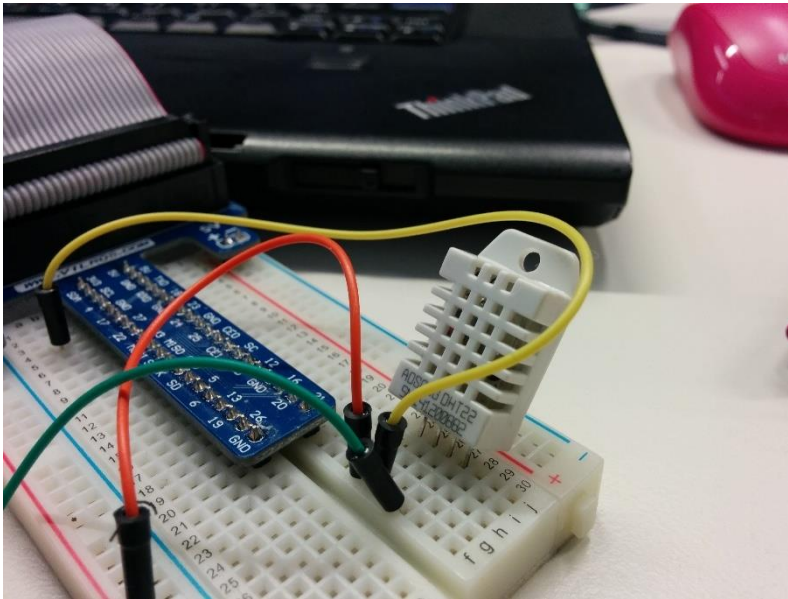
El circuito es muy sencillo, el sensor de temperatura y humedad DHT22 tiene 4 patillas de las que se utilizan sólo 3:

DHT22 pins	
1	VCC
2	DATA
3	NC
4	GND

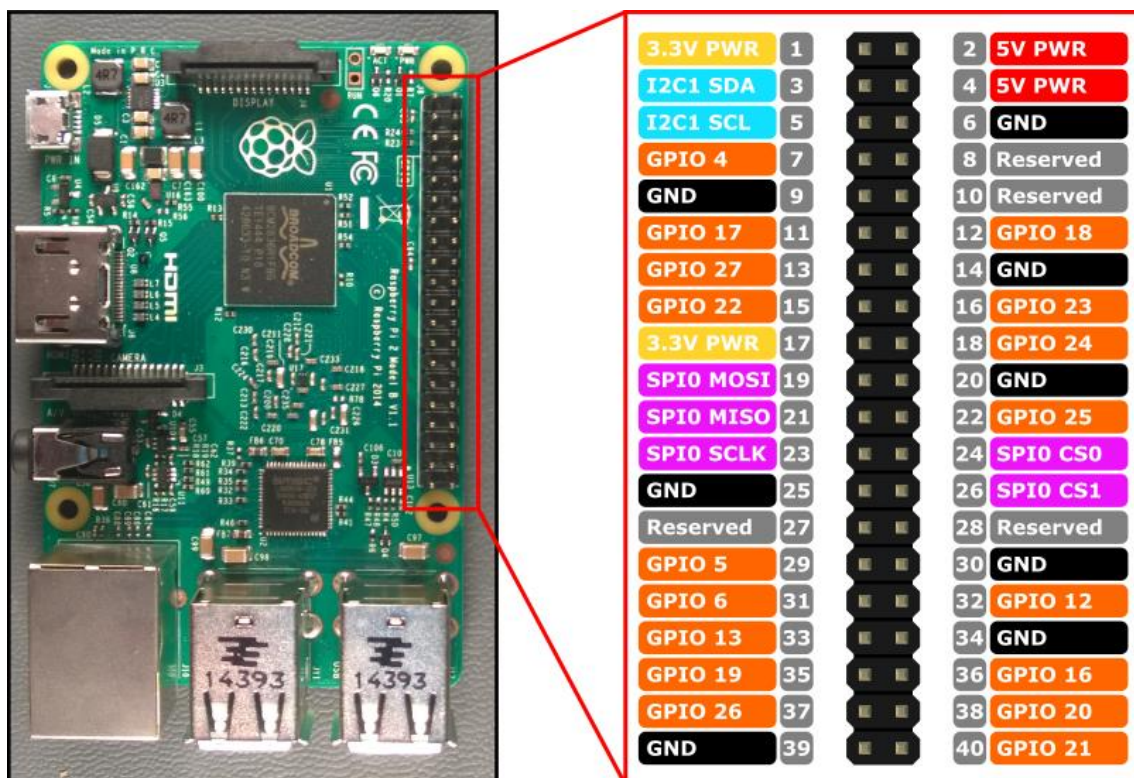


La patilla 1 se conecta a 3.3 (+) voltios, la patilla 4 a toma de tierra (-) y la patilla 2, que es la que proporciona los datos, la conectaremos al pin 4 de la Raspberry Pi 2. Si estamos utilizando un conector T-Cobbler con el cable Breakout, en la T podremos ver que están marcados los pines.





En el caso de utilizar la Raspberry Pi 2 directamente, aquí está el pinout:



Paso 2: Creación de recursos en Azure

Antes de empezar con el código de la Raspberry Pi 2, necesitaremos algunos recursos en Azure para conectar nuestro dispositivo y gestionar la transmisión de datos:

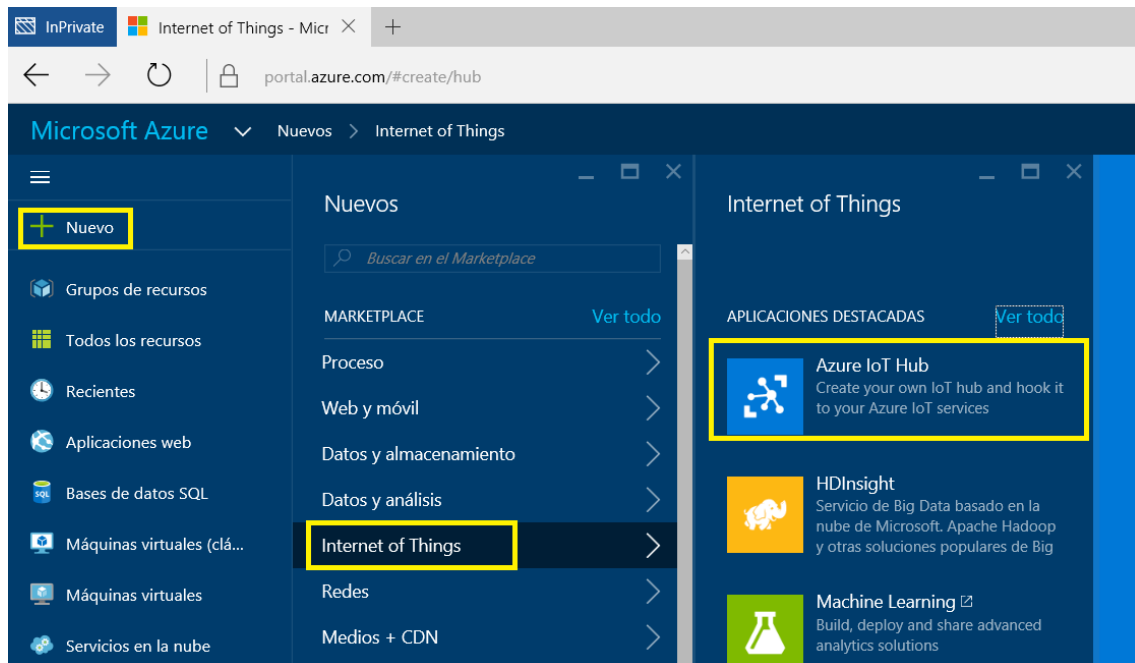
- **IoT Hub:** encargado de la gestión de dispositivos de IoT, incluida la ingesta de datos.
- **Stream Analytics:** encargado del análisis de los datos.

- **Event Hub:** ingesta de datos para dispositivos no gestionados y otras fuentes como Stream Analytics

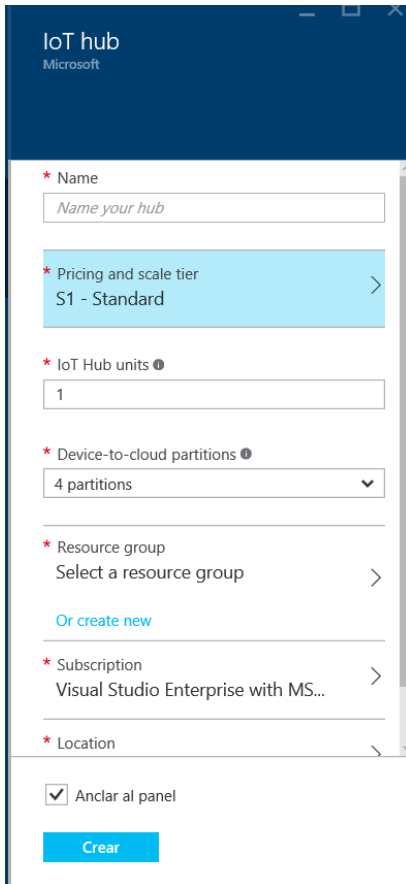
Creación del IoT Hub

Para crear el IoT Hub debemos acceder al portal de Azure (<http://portal.azure.com>).

Pulsamos sobre el botón “Nuevo”, luego seleccionamos “Internet of Things” y a continuación “Azure IoT Hub”.



Accedemos al asistente de creación del IoT Hub.



Le damos un nombre y en “Resource Group” seleccionamos la opción de crear uno nuevo. Un grupo de recursos es una agrupación de servicios de Azure que se gestiona de forma conjunta. Todos los servicios que demos de alta durante el taller los meteremos dentro del mismo grupo de recursos.

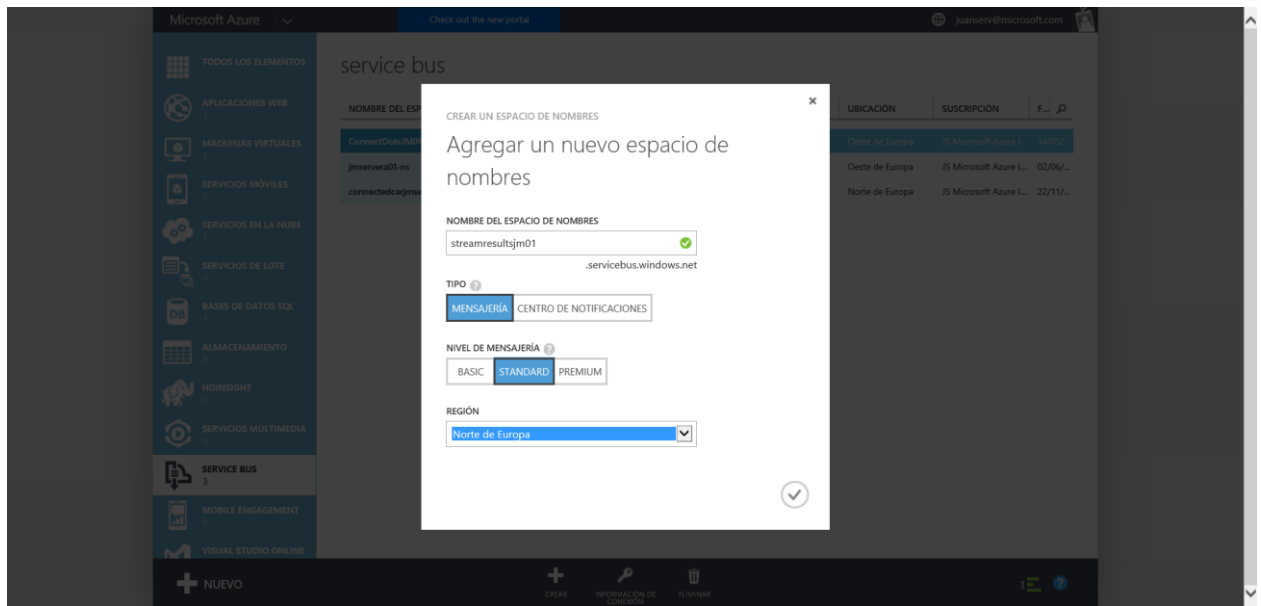
En “Location” vamos a seleccionar “North Europe”. Este campo representa el datacenter en el que vamos a almacenar nuestros datos, en nuestro caso elegimos norte de Europa por la cercanía, pero podríamos coger cualquier datacenter del mundo.

Pulsamos en crear y veremos en la pantalla de inicio que se crea un acceso directo. El proceso de crear el IoT Hub puede tardar unos minutos.

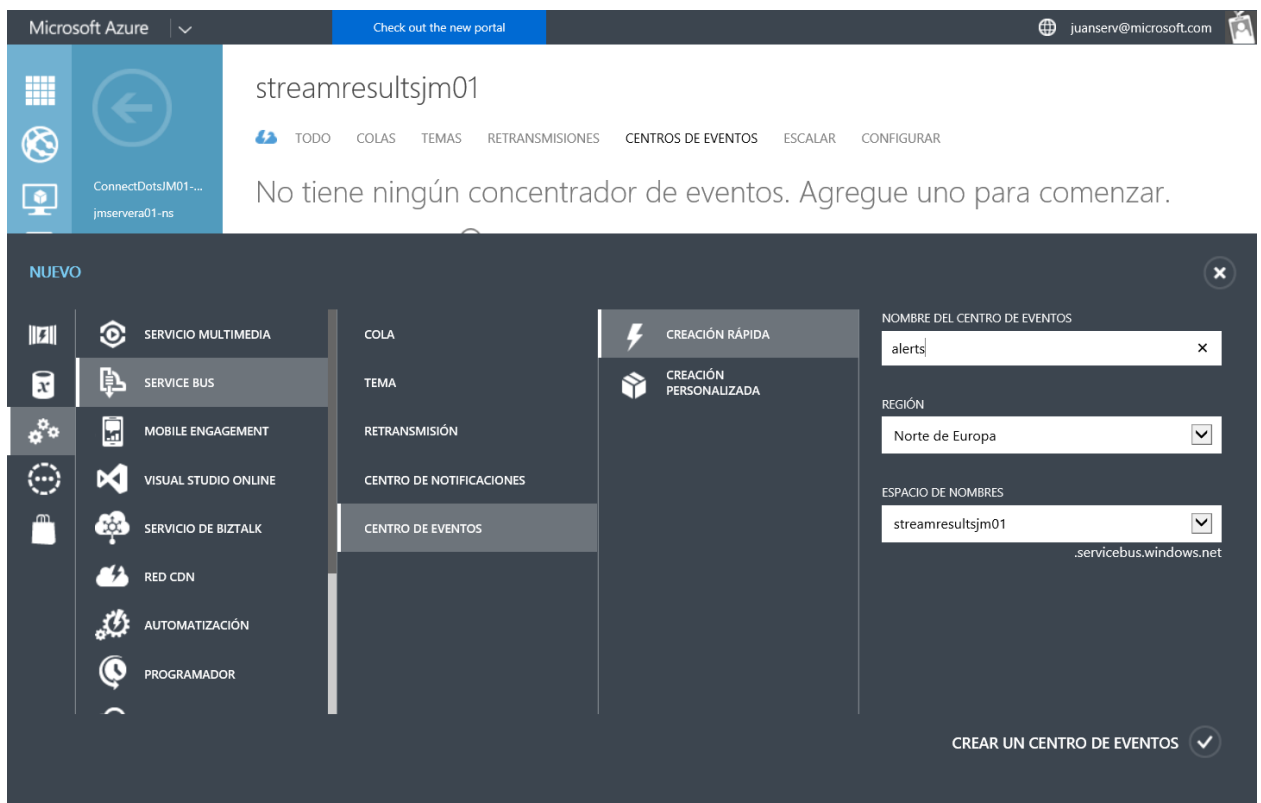
Creación de Event Hubs

Event Hubs es una cola de ingestión de mensajes muy potente que escala a millones de eventos por segundo. Vamos a utilizar esta cola como salida del análisis de datos que realizaremos con Stream Analytics.

El Centro de Eventos (Event Hub) en Service Bus, por ahora sólo está en el portal antiguo en <https://manage.windowsazure.com>. Primero debemos crear un espacio de nombres en Service Bus con tipo “Mensajería”:



Una vez creado el espacio de nombres, ya podemos crear un Centro de Eventos (Event Hub), lo llamaremos alerts:






Para poder utilizar el centro de eventos necesitaremos una clave de acceso, primero deberemos crear una directiva de acceso compartido (SAS), a través de la configuración, la hemos llamado “fullaccess”:

alerts

PANEL CONFIGURAR GRUPOS DE CONSUMIDORES

general

RETENCIÓN DE MENSAJES	<input type="text" value="1"/>	días	
ESTADO DEL CONCENTRADOR DE EVENTOS	<input type="button" value="Habilitado"/>		
RECuento DE PARTICIONES	<input type="text" value="4"/>	Particiones	

directivas de acceso compartido

NOMBRE	PERMISOS
<input type="text" value="fullaccess"/>	<input type="button" value="Administrar, Enviar, Escuchar"/>
<input type="text" value="NUEVO NOMBRE DE DIRECTIVA"/>	<input checked="" type="checkbox"/> Administrar <input checked="" type="checkbox"/> Enviar <input checked="" type="checkbox"/> Escuchar

 GUARDAR
  DESCARTAR
 1  

Está creada con permisos completos (Administrar). Una vez creada, en la sección Panel nos aparece una llave (conexiones) que nos permitirá obtener la cadena de conexión para utilizarla luego:

Información de conexión de acceso

Utilice esta información de conexión para conectarse al centro de eventos "alerts".

SAS 	
NOMBRE	CADENA DE CONEXIÓN
fullaccess	Endpoint=sb://streamresultsjm01.servicebus.windows.net;/SharedAccessKeyName=full



Más adelante también necesitaremos la conexión a nivel de Service Bus para poder gestionar los grupos de consumo:

Información de conexión de acceso

X

Use esta información de conexión para administrar el espacio de nombres "streamresultsjm01". También puede usar directivas de autorización configuradas para conectarse a todas las entidades de este espacio de nombres.

SAS ?

NOMBRE	CADENA DE CONEXIÓN	
RootManageSharedAccessKey	Endpoint=sb://streamresultsjm01.servicebus.windows.net;/SharedAccessKeyName=Ro	

ACS

¿Está buscando información acerca de la conexión ACS? Consulte este [vínculo](#) para obtener más información acerca del uso de ACS con Bus de servicio.

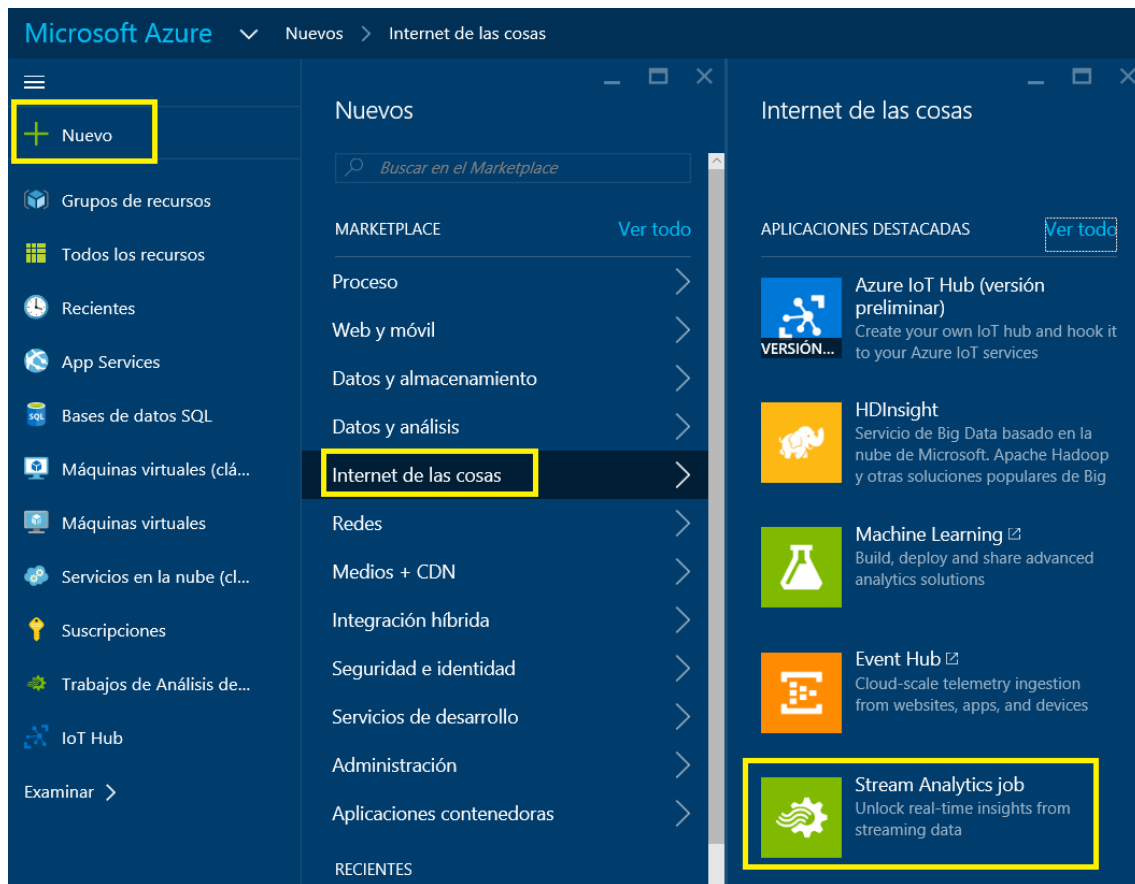
Creación de Stream Analytics

Hasta ahora hemos creado la entrada y la salida de lo que va a ser el análisis de datos en tiempo real, un IoT Hub para la gestión del dispositivo y la recepción de datos y un Event Hub para recibir la salida del stream.

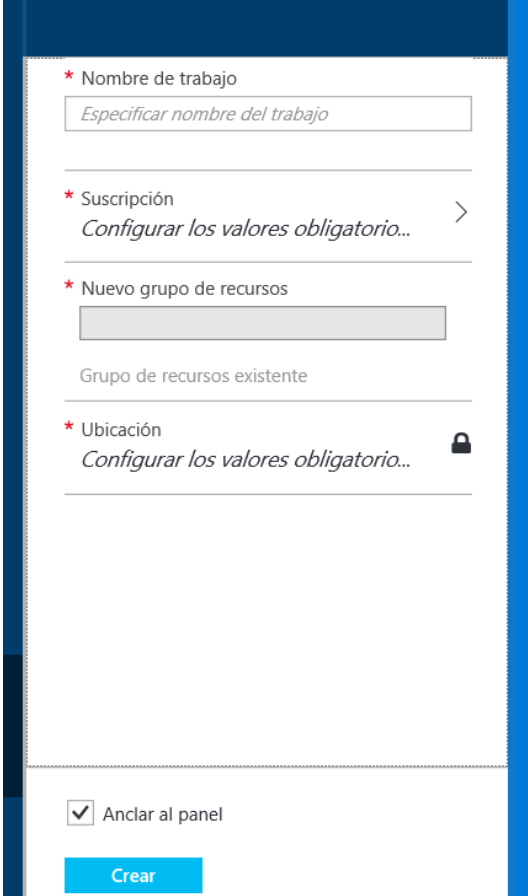
Ahora vamos a crear un trabajo de Stream Analytics que nos permitirá coger los datos del IoT Hub hacer una consulta sobre ellos y mostrar el resultado en un sitio web o en un panel de mandos. En este caso vamos a generar una alerta cuando la temperatura exceda un determinado nivel:

Para esto usaremos el portal de Azure (<http://portal.azure.com>).

Pulsamos en “Nuevo”, “Internet de las cosas”, “Stream Analytics Job”.



Se abrirá el asistente de creación de trabajos de Stream Analytics.



The screenshot shows a configuration form for creating a new IoT Hub. The form is titled "Nombre de trabajo" (Work name) and includes several sections:

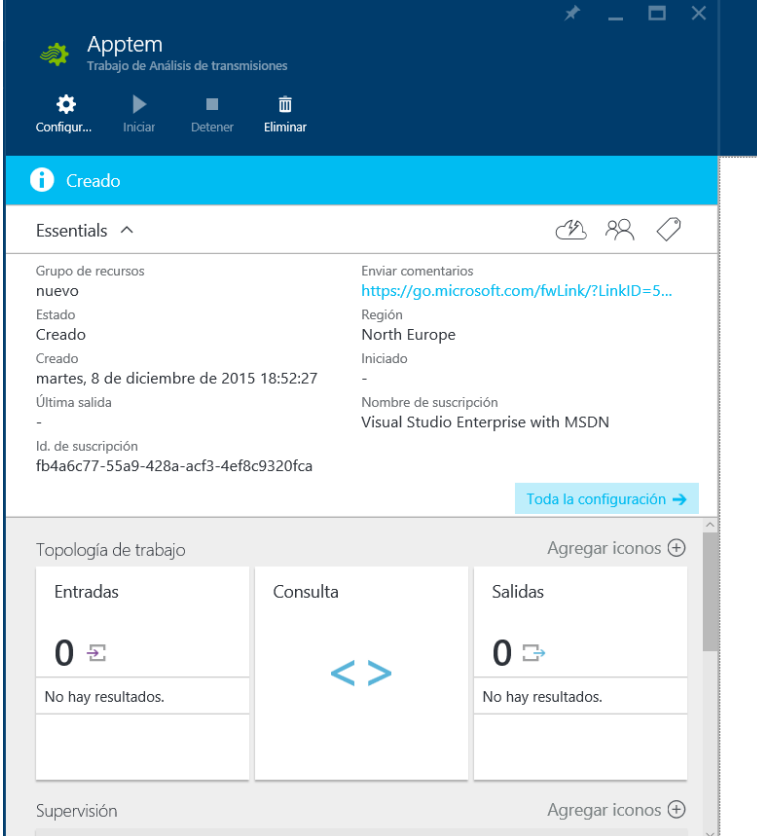
- Nombre de trabajo**: A text input field with the placeholder "Especificar nombre del trabajo".
- Suscripción**: A section with a right arrow icon and the text "Configurar los valores obligatorio..." (Configure required values...).
- Nuevo grupo de recursos**: A text input field.
- Grupo de recursos existente**: A section with a right arrow icon and the text "Configurar los valores obligatorio..." (Configure required values...).
- Ubicación**: A section with a lock icon and the text "Configurar los valores obligatorio..." (Configure required values...).

At the bottom of the form, there is a checkbox labeled "Anclar al panel" (Pin to panel) which is checked. Below the checkbox is a blue button labeled "Crear" (Create).

Le asignamos un nombre al trabajo. En la ubicación elegimos la misma que hemos seleccionado al crear el IoT Hub (Norte de Europa). En el grupo de recursos seleccionamos el que hemos creado para el IoT Hub.

Si vamos al apartado de "Todos los elementos" veremos que se nos ha creado el trabajo de Stream Analytics.

Accedemos a él para configurar las entradas, las salidas y la consulta que se va a hacer sobre los datos.



Apptem
Trabajo de Análisis de transmisiones

Configur... Iniciar Detener Eliminar

Creado

Essentials ^

Grupo de recursos
nuevo
Estado
Creado
Creado
martes, 8 de diciembre de 2015 18:52:27
Última salida
-
Id. de suscripción
fb4a6c77-55a9-428a-acf3-4ef8c9320fca

Enviar comentarios
<https://go.microsoft.com/fwlink/?LinkID=5...>
Región
North Europe
Iniciado
-
Nombre de suscripción
Visual Studio Enterprise with MSDN

Toda la configuración →

Topología de trabajo

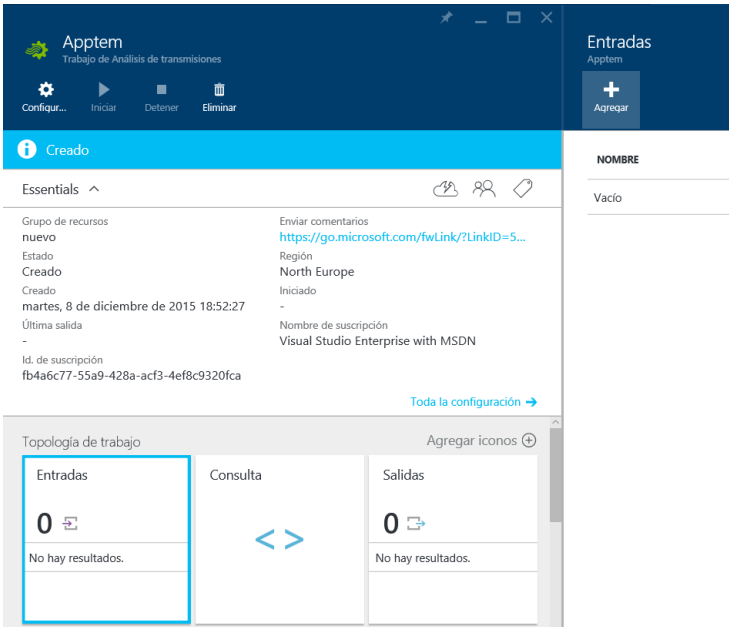
Entradas
0
No hay resultados.

Consulta
<>

Salidas
0
No hay resultados.

Supervisión

Vamos a introducir primero la entrada. Al pulsar sobre “Entradas” podremos configurarlas:



Apptem
Trabajo de Análisis de transmisiones

Configur... Iniciar Detener Eliminar

Entradas
Apptem
+ Agregar

Creado

Essentials ^

Grupo de recursos
nuevo
Estado
Creado
Creado
martes, 8 de diciembre de 2015 18:52:27
Última salida
-
Id. de suscripción
fb4a6c77-55a9-428a-acf3-4ef8c9320fca

Enviar comentarios
<https://go.microsoft.com/fwlink/?LinkID=5...>
Región
North Europe
Iniciado
-
Nombre de suscripción
Visual Studio Enterprise with MSDN

Toda la configuración →

Topología de trabajo

Entradas
0
No hay resultados.

Consulta
<>

Salidas
0
No hay resultados.

Supervisión

Nueva entrada

* Alias de entrada

* Tipo de origen ⓘ

Flujo de datos

* Origen ⓘ

Centro de eventos

* Espacio de nombres de bus de servicio ⓘ

* Nombre del centro de eventos ⓘ

* Nombre de directiva de centro de eventos ⓘ

* Clave de directiva de centro de eventos ⓘ

Crear

En el asistente introducimos un nombre para la entrada. Seleccionamos “Flujo de datos”, “Centro de IoT” y le damos un nombre. Seleccionamos el nombre del IoT Hub que tengamos creado en la suscripción y ponemos el nombre de la política de acceso (iothubowner) y la cadena de conexión (se encuentra en el IoT Hub en la parte de políticas de acceso en iothubowner). Guardamos la entrada y nos vamos al parte de “Consulta”.

Microsoft Azure

Apptem > Apptem

Nuevo

Grupos de recursos

Todos los recursos

Recientes

App Services

Bases de datos SQL

Máquinas virtuales (clá...

Apptem

Consulta

Guardar

Descartar

1 SELECT

2 *

3 INTO

4 [YourOutputAlias]

5 FROM

6 [YourInputAlias]

13

Borramos el contenido de la consulta e introducimos el siguiente:

```
SELECT
    'TempSpike' AS alerttype,
    'Temperature over 25C' AS message,
    displayname,
    guid,
    measurename,
    unitofmeasure,
    location,
    organization,
    MIN(timecreated) AS timecreated,
    MAX(value) AS tempMax,
    MAX(value) AS value
FROM
    Input TIMESTAMP BY timecreated
WHERE
    measurename = 'temperature' OR measurename = 'Temperature'
GROUP BY
    displayname, guid, measurename, unitofmeasure, location,
    organization,
    TumblingWindow(Second, 5)
HAVING
    tempMax > 25
```

En la cláusula From hemos de poner el nombre que le hayamos dado a la entrada de la consulta (en nuestro caso Input).

Una vez creada la consulta vamos a definir la salida del flujo:

Nueva salida

* Alias de salida

* Receptor ⓘ

Centro de eventos ▼

* Espacio de nombres de bus de servicio ⓘ

* Nombre del centro de eventos ⓘ

* Nombre de directiva de centro de eventos ⓘ

* Clave de directiva de centro de eventos ⓘ

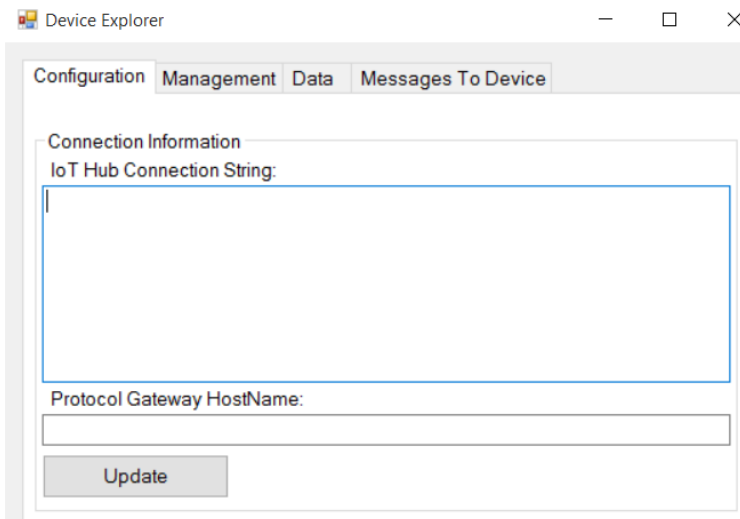
Columna de clave de partición ⓘ

Crear

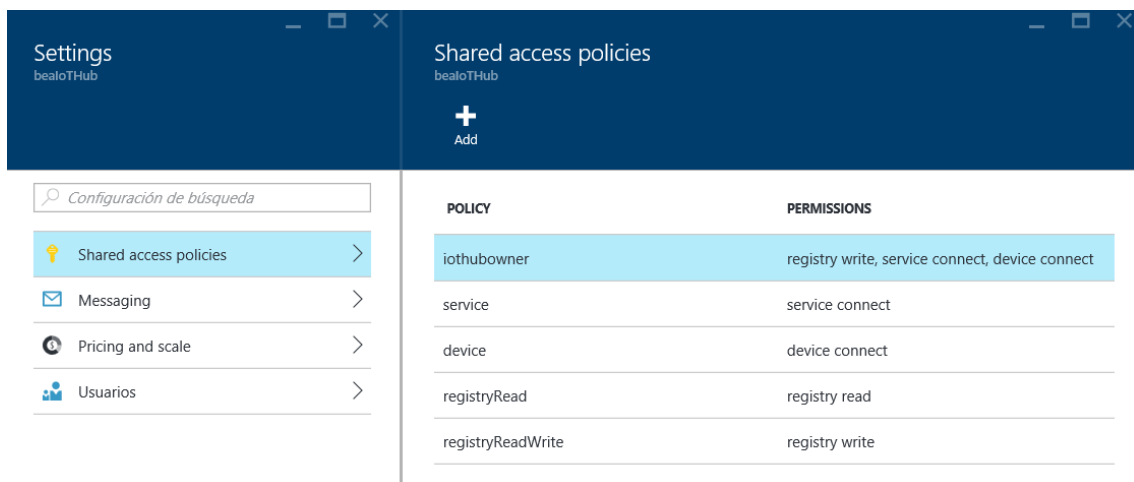
La salida será a un Centro de Eventos, tendremos que extraer la información del centro de eventos.

Paso 3: Registro del dispositivo

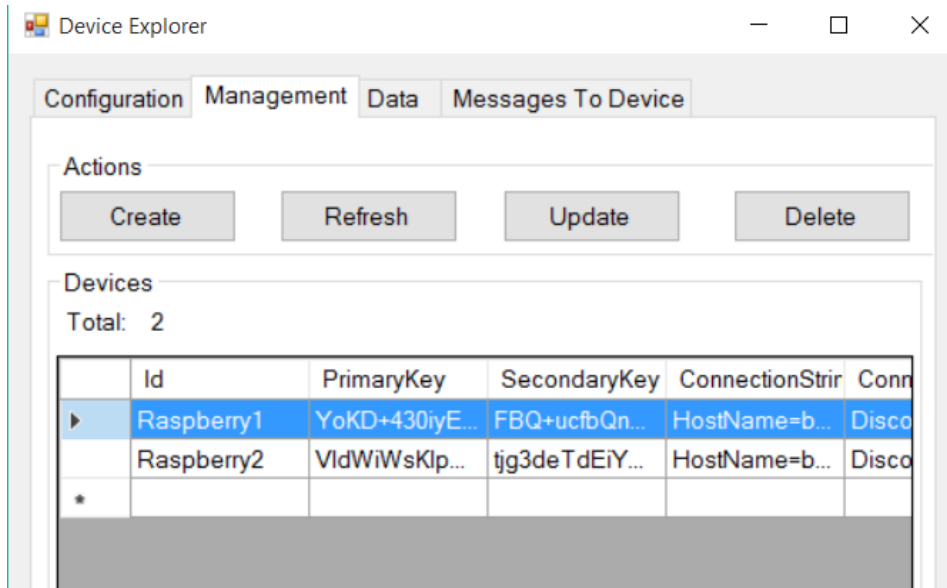
IoT Hub realiza conexiones individualizadas y bidireccionales con cada dispositivo, de forma que en el futuro podamos gestionar de forma segura todos los dispositivos conectados al Hub. Por esto necesitamos registrar el dispositivo, para lo que usaremos una herramienta llamada [Device Explorer](#). Esta herramienta utiliza la API de IoT Hub para realizar el registro y sirve de ejemplo para crearnos nuestra propia herramienta de gestión de dispositivos, pero para este taller usaremos directamente la herramienta de ejemplo.



Lo primero que hemos de hacer es introducir la cadena de conexión de nuestro IoT Hub. Para saber cuál es, vamos al portal de Azure y seleccionamos el IoT Hub. En la parte de políticas de acceso seleccionamos la de “iothubowner” y ahí encontraremos la cadena de conexión. Es importante utilizar esta política, pues las demás no tienen nivel de acceso suficiente para realizar todas las acciones que puede ejecutar la herramienta.



La copiamos y la ponemos en el device explorer, pulsamos en “Update”. Una vez hemos establecido esta conexión con el IoT Hub, debemos crear un dispositivo. En la pestaña de “Management” pulsamos en “Create” y le asignamos un nombre al dispositivo. Tanto el nombre del dispositivo como la clave que se le asignará las tendremos que meter en el código de nuestra aplicación que vayamos a desplegar en la Raspberry. De esta manera la Raspberry ya quedará identificada y se podrá comunicar con el IoT Hub.



Paso 4: Código de Windows 10 IoT Core

El código para la lectura del sensor y comunicación con el IoT Hub está disponible en Git Hub (<https://github.com/jmservera/iot/tree/vs2015update1>). En el código encontraremos 3 proyectos:

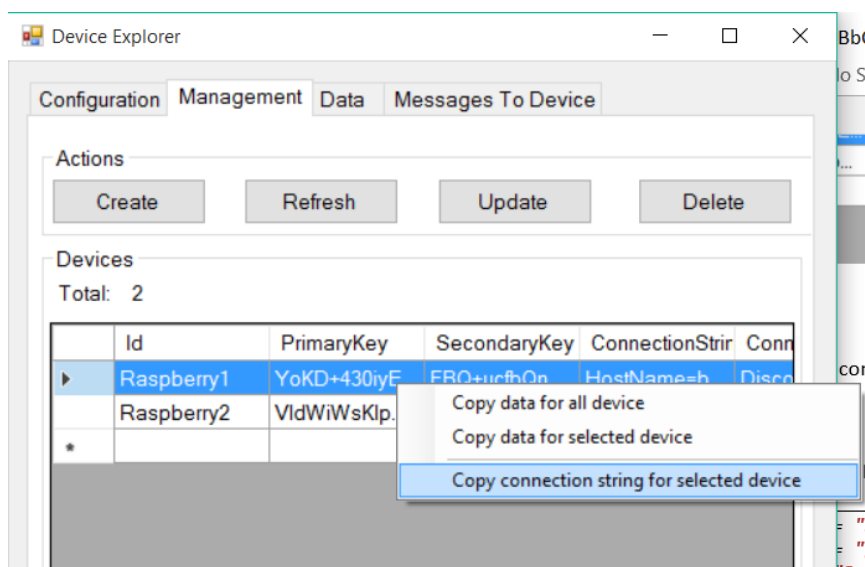
- IoTensors: el proyecto que utilizaremos ahora, contiene el código para leer el sensor DHT22 y conectar al IoT Hub desde una aplicación universal que desplegaremos en la Raspberry Pi 2
- SensorTag: un proyecto de prueba del sensor CS2650 de Texas Instruments
- Website: una copia adaptada del website que encontrarás también en <http://connectthedots.io>

Abrimos la clase MainPage.xaml.cs del proyecto DhtView, donde introduciremos los valores que hemos obtenido

```
const string iotHubUri = "[nombre].azure-devices.net";
const string deviceKey = "[clave]";
const string deviceName = "[nombre de dispositivo]";
const string GUID = "[crear un guid]";
```

En "iotHubUri" ponemos la primera parte de la cadena de conexión del IoT Hub.

En "devicekey" ponemos la clave del dispositivo. Para saber cuál es la clave, desde Device Explorer pulsamos el botón derecho sobre el dispositivo y seleccionamos la opción de copiar la cadena de conexión del dispositivo.

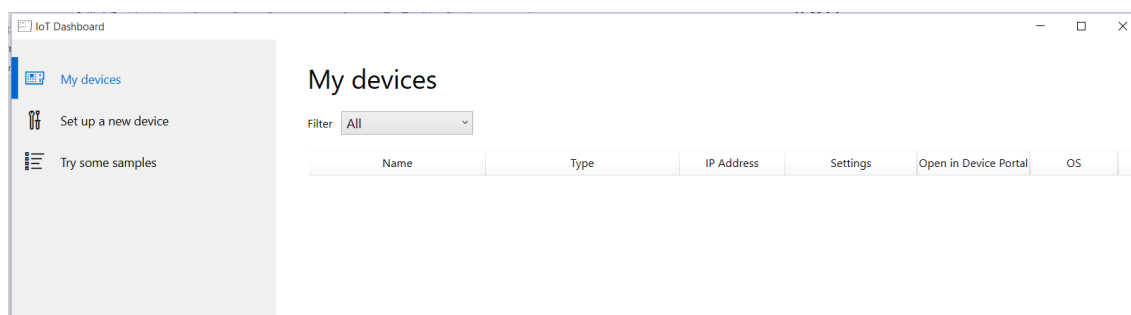


Cogemos la parte de SharedAccessKey y la introducimos en “devicekey”.

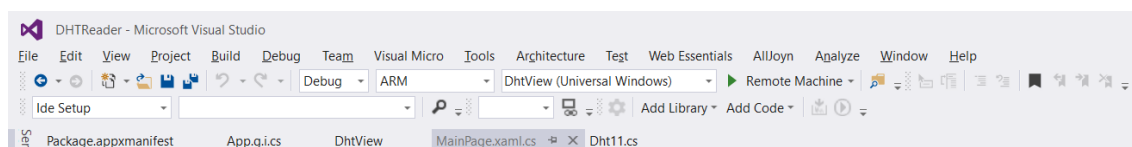
En “deviceName” ponemos el nombre que le hemos dado al dispositivo.

Despliegue del código en la Raspberry Pi 2

Una vez hemos modificado el código lo vamos a desplegar en la Raspberry Pi 2. Para ello necesitamos conocer la IP del dispositivo. Usaremos la aplicación “Windows 10 IoT Core Dashboard” para saber la IP. Abrimos la aplicación y en la parte My Devices buscamos nuestro dispositivo (si hay más dispositivos en la red intentamos identificarlo por el nombre).

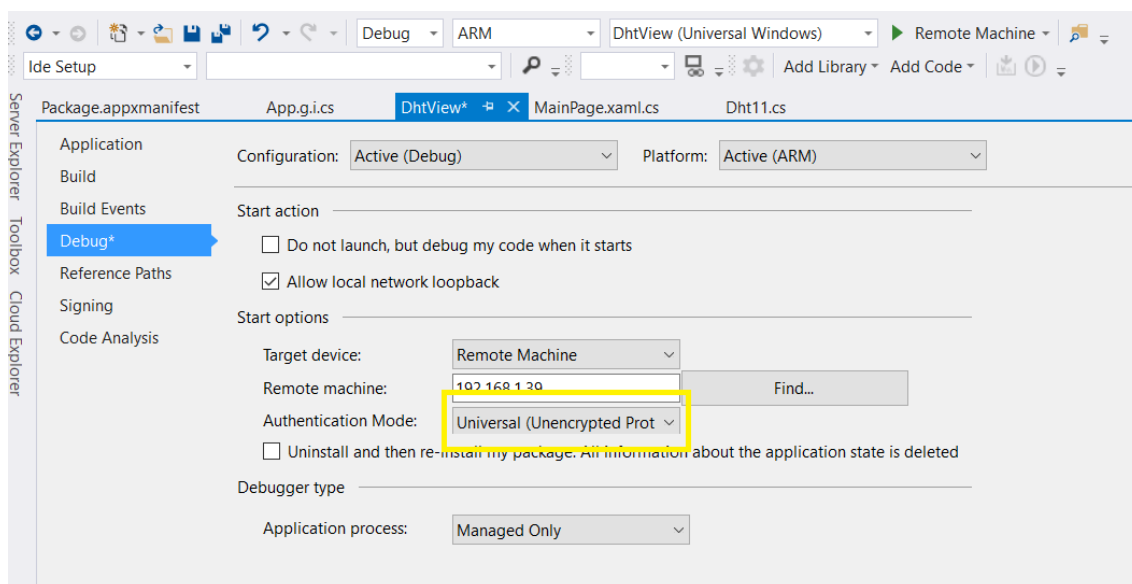


Una vez tengamos la IP, configuramos Visual Studio en modo ARM (la tecnología de procesador que utiliza la Raspberry Pi 2) y en el modo de ejecución ponemos máquina remota.



Se abrirá una ventana en la que nos pedirá que introduzcamos la IP. A continuación, antes de ejecutar, debemos asegurarnos que la autenticación está desactivada o en modo Universal.

Para ello accedemos a las propiedades del proyecto y en el apartado Debug en el modo de autenticación seleccionamos None.



Guardamos y ya podemos desplegar el proyecto, simplemente ejecutamos y Visual Studio instalará todo lo necesario en la Raspberry durante el despliegue.

Podremos ver en el código que se realiza una conexión a IoT Hub y se crean dos tareas, una para enviar eventos y otra para recibirlos desde IoT Hub:

```
private async void InitializeSensor()
{
    var key =
AuthenticationMethodFactory.CreateAuthenticationWithRegistrySymmetricKey(deviceName, deviceKey);
    DeviceClient deviceClient = DeviceClient.Create(iotHubUri, key,
TransportType.Http1);

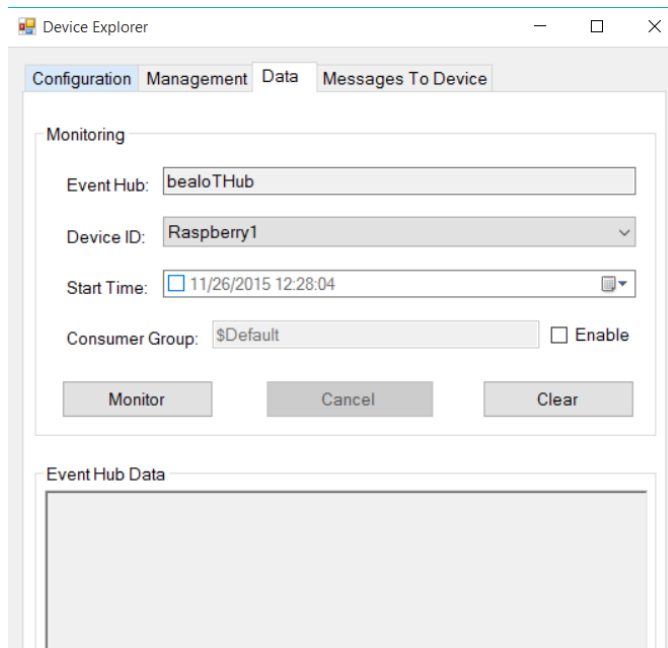
    Task ts = SendEvents(deviceClient);
    Task tr = ReceiveCommands(deviceClient);

    await Task.WhenAll(ts, tr);
}
```

Paso 5: monitorización

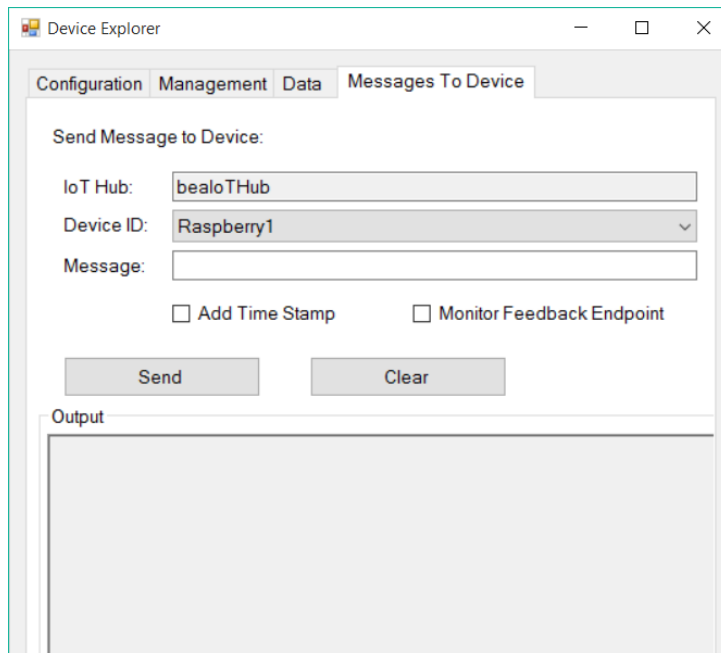
Una vez el código haya cargado y comience a ejecutarse podemos monitorizar las lecturas que se están haciendo de temperatura y humedad y podemos enviar mensajes.

Desde Device Explorer podemos ver lo que se está enviando desde la Raspberry al IoT Hub en la pestaña de Data.



Pulsamos en “Monitor” y empezaremos a ver cómo aparecen las lecturas de humedad y temperatura.

En el apartado de Messages To Device podemos enviar mensajes al dispositivo y ver cómo se reciben a través de la salida por consola de la aplicación.



Si tuviéramos una pantalla conectada a la Raspberry Pi 2 también podríamos ver todo el resultado por pantalla, pues la aplicación tiene definido un interfaz de usuario.

Paso 6a: Power BI

Para poder ver las lecturas del sensor en tiempo real desde Power Bi vamos a crear otra consulta de Stream Analytics.

Creemos el trabajo igual que hemos hecho en el apartado 2.

Como entrada seleccionamos lo mismo que en el apartado 2 (el IoT Hub).

En cuanto a la consulta, borramos el contenido de la consulta e introducimos el siguiente:

```
Select measurename,  
Displayname,  
timecreated,  
Value,  
From Input
```

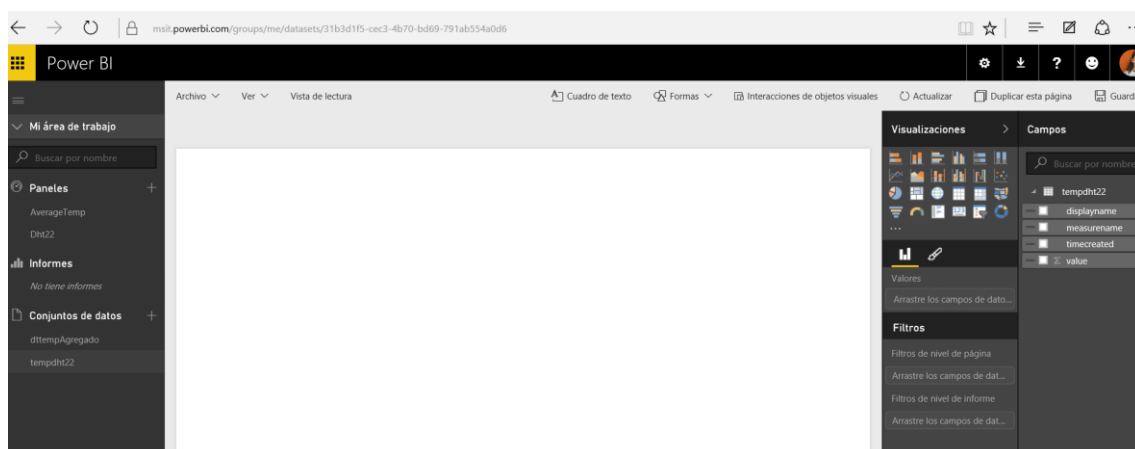
En la cláusula From hemos de poner el nombre que le hayamos dado a la entrada de la consulta.

Vamos a la parte de Salidas.

En el asistente de creación de salida seleccionamos “Power BI”, nos va a pedir que iniciemos sesión o que creamos una cuenta nueva (es gratuito, pero se necesita una cuenta corporativa). Una vez establecida la conexión introducimos nombre para el dataset (conjunto de resultados que van a pasar a Power BI), este será el nombre que aparecerá en PowerBI.

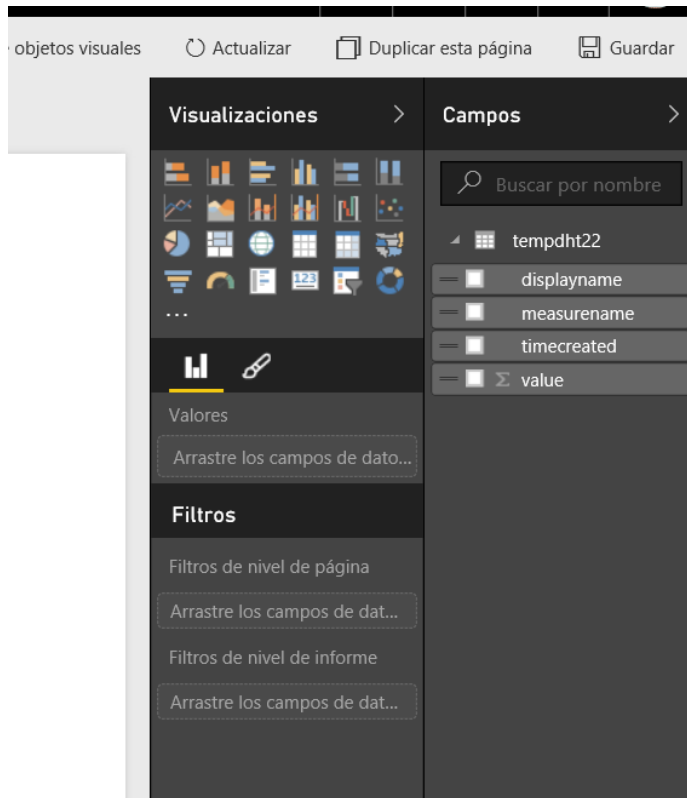
Debemos iniciar el trabajo de Stream Analytics para que empiece a escuchar si hay datos en el IoT Hub.

Abrimos el portal de Power BI (<http://powerbi.microsoft.com>).

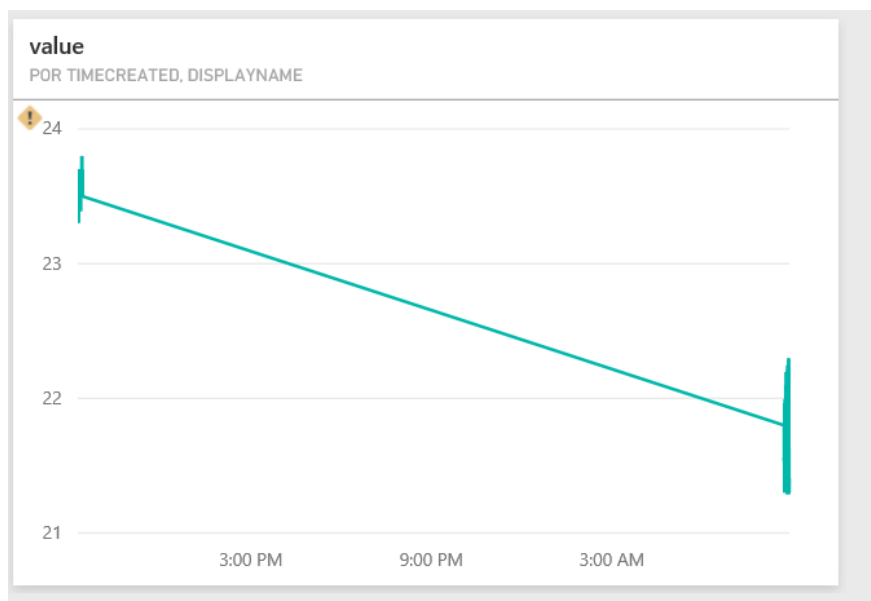


Vemos que en el apartado de Conjuntos de Datos (datasets), se ha creado uno con el nombre que hemos puesto en la salida de StreamAnalytics.

Lo seleccionamos y vemos que en la parte derecha tenemos todas las opciones para generar gráficos y seleccionar los datos que queremos que aparezcan.



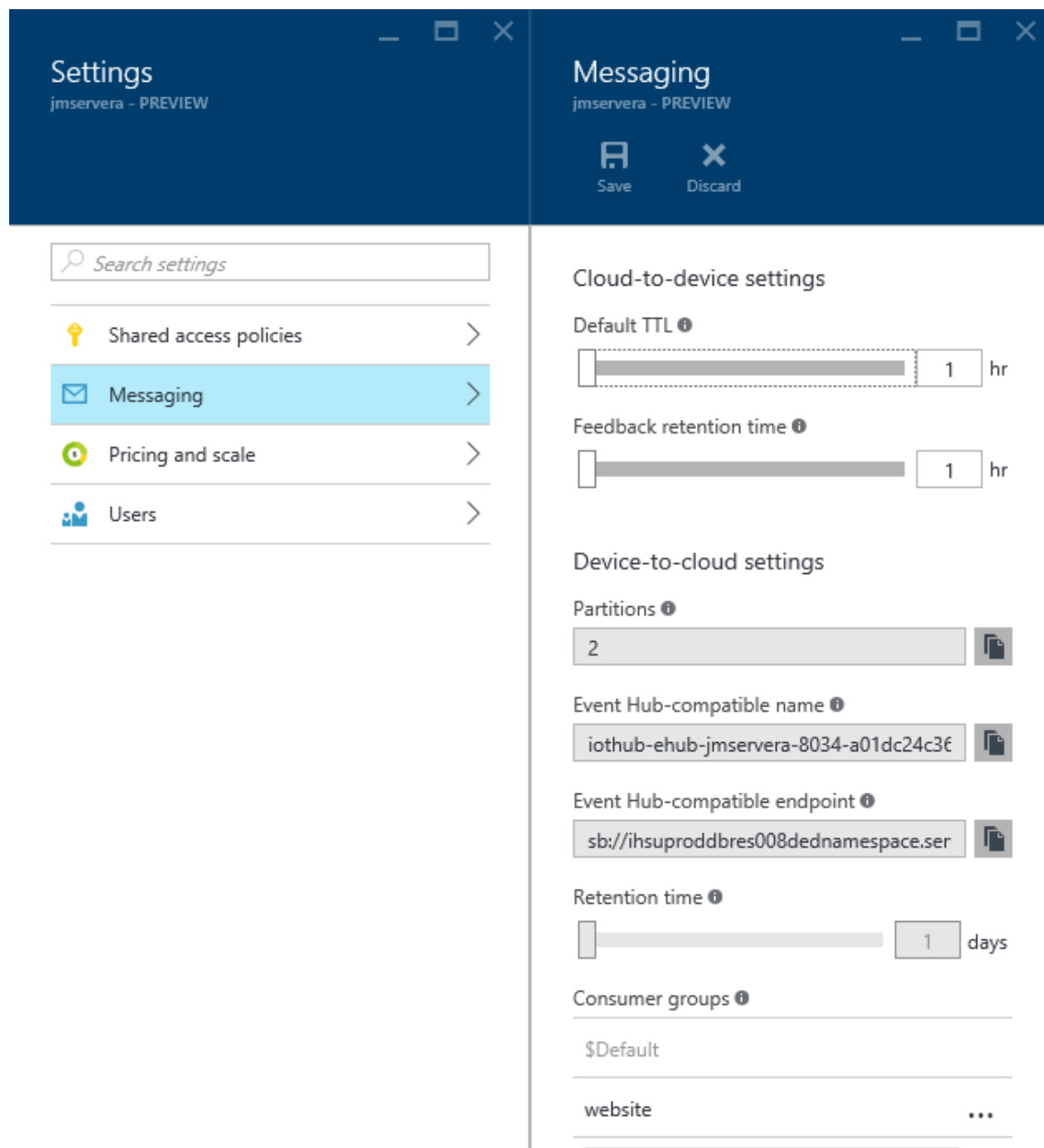
Desde aquí podemos crear informes y ponerlos luego en la pantalla de inicio.



En tiempo real se irán mostrando los valores de las lecturas que hayamos seleccionado al crear el informe.

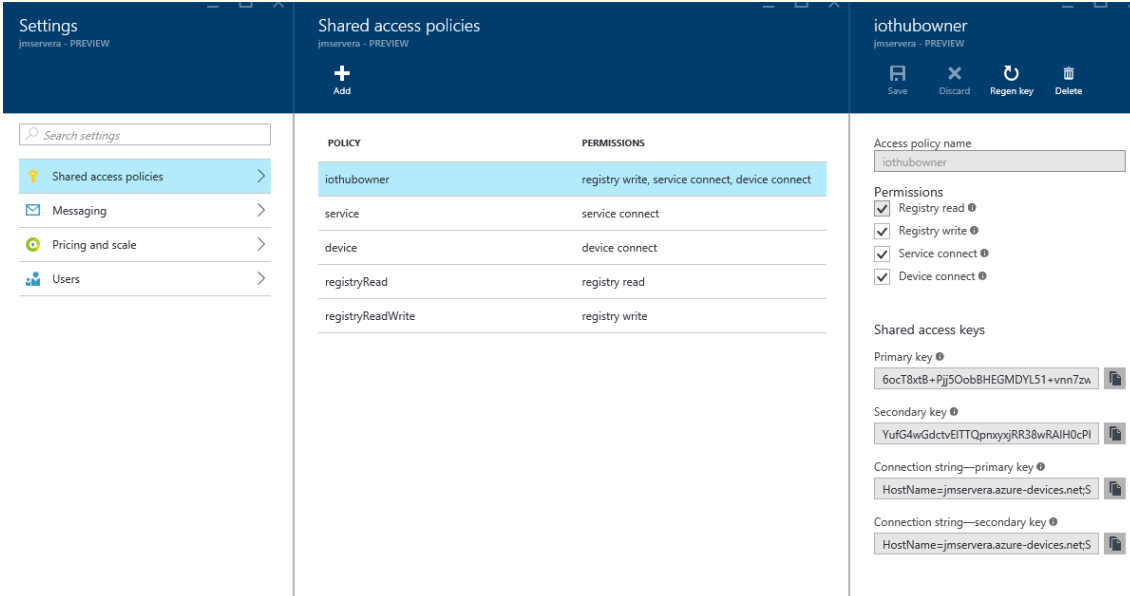
Paso 6b: Website

Si no tenemos una cuenta de PowerBI podemos montar un WebSite en Azure que recibirá los datos. Para ello en la carpeta website de <https://github.com/jmservera/iot/tree/vs2015update1> tenemos una aplicación Web para Visual Studio 2015. Deberemos obtener unos datos del IoT Hub para conectar desde el website. El primero es el string de conexión compatible con EventHubs:



Fijaos que además hay que crear un consumer Group, en este caso lo hemos llamado “website”, también crearemos otro llamado “local” que nos permitirá hacer pruebas en local.

Y lo tendremos que completar con la clave que nos aparece en la Shared Access Policies:



POLICY	PERMISSIONS
iothubowner	registry write, service connect, device connect
service	service connect
device	device connect
registryRead	registry read
registryReadWrite	registry write

Access policy name: iothubowner

Permissions:

- ☒ Registry read
- ☒ Registry write
- ☒ Service connect
- ☒ Device connect

Shared access keys:

Primary key: 6ocT8xtB+Pj5OobBHEGMDYL51+vmn7Zw

Secondary key: YufG4wGdctvEITTQpnxyjRR38wRAIH0cPI

Connection string—primary key: HostName=jmservice.azure-devices.net;S

Connection string—secondary key: HostName=jmservice.azure-devices.net;S

Además necesitaremos una cuenta de almacenamiento. En el archivo web.config del sitio web buscaremos las entradas `Microsoft.ServiceBus.EventHubAlerts` y siguientes, sustituiremos los valores con los valores que hemos obtenido en el portal. Necesitaremos combinar el connection string que encontramos en “Shared acces policies” con los valores que encontraremos en la sección Messaging para compatibilidad con Event Hubs.

También necesitaremos los valores del EventHub que hemos creado como “alerts” para poder visualizar las alertas en la web y el string de conexión general de Service Bus (son dos diferentes).

Así en EventHubDevices/ConnectionStringDevices utilizaremos los valores del IoT Hub y en EventHubAlerts/ConnectionStringAlerts los del Event Hub.

```
<add key="Microsoft.ServiceBus.EventHubAlerts" value="alerts"/>
<add key="Microsoft.ServiceBus.ConnectionStringAlerts"
value="Endpoint=sb://streamresultsjm01.servicebus.windows.net/;SharedAccessKeyName=fullaccess;SharedAccessKey=xxxxxx"/>

<add key="Microsoft.ServiceBus.EventHubDevices" value="iothub-ehub-jmservice-8034-a01dc24c36"/>
<add key="Microsoft.ServiceBus.ConnectionStringDevices"
value="Endpoint=sb://ihsuproddbres008dednamespace.servicebus.windows.net/;HostName=jmservice.azure-devices.net;SharedAccessKeyName=iothubowner;SharedAccessKey=xxxxx"/>

<add key="Microsoft.ServiceBus.ConnectionString"
value="Endpoint=sb://streamresultsjm01.servicebus.windows.net/;SharedAccessKeyName=RootManageSharedAccessKey;SharedAccessKey=xxxxxx"/>
<add key="Microsoft.Storage.ConnectionString"
value="DefaultEndpointsProtocol=https;AccountName=connectdotsjm01storage;AccountKey=xxxxxx"/>
```

Una vez sustituidos, podremos ejecutar el website y ver si cómo están llegando los valores. El Website utiliza websockets para enviar al cliente

Paso 7: Retos

- Utilizar más elementos de Stream Analytics, podemos crear otros análisis de datos con alertas para la humedad con Joins temporales:

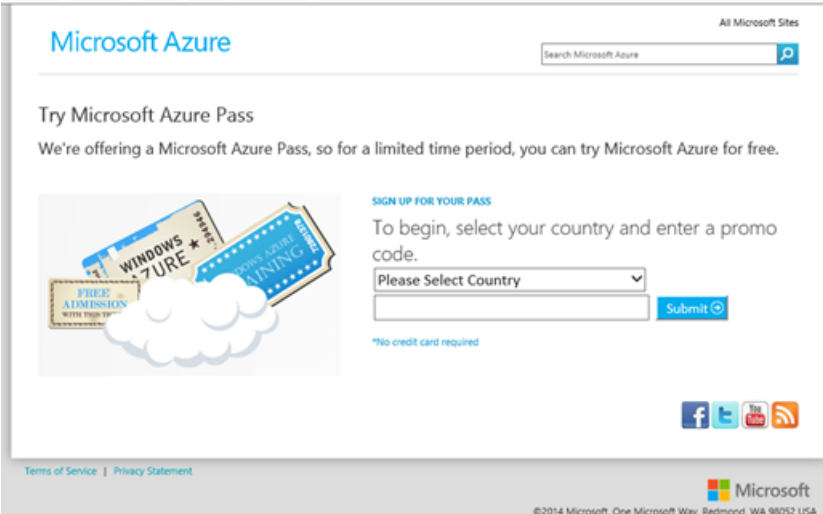
<https://github.com/jmservera/connectthedots/tree/master/Azure/StreamAnalyticsQueries>

- Utilizar Machine Learning para detección de anomalías:

<https://github.com/Azure/connectthedots/tree/master/Azure/MachineLearning>

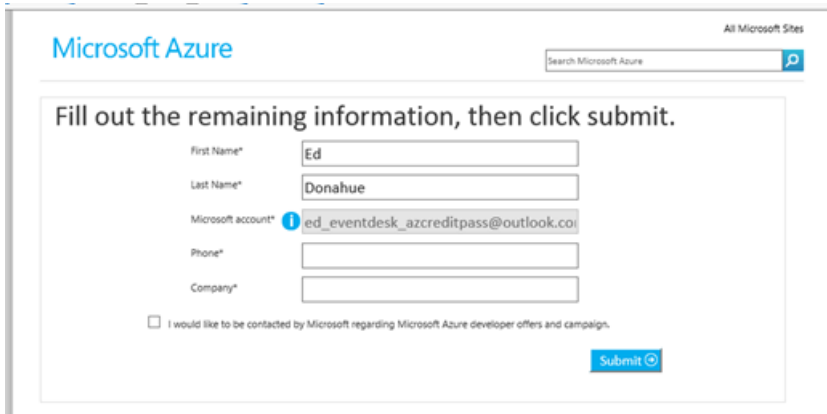
Anexo: activación Azure Pass

1. Ir a <http://microsoftazurepass.com>. Selecciona el país, escribe el código que te ha sido proporcionado y pincha en Submit.



2. Inicia sesión con una cuenta de Microsoft para poder continuar.

3. Completa la información y pulsa en Submit.



Microsoft Azure

Search Microsoft Azure

Fill out the remaining information, then click submit.

First Name* Ed

Last Name* Donahue

Microsoft account* ed_eventdesk_azcreditpass@outlook.co

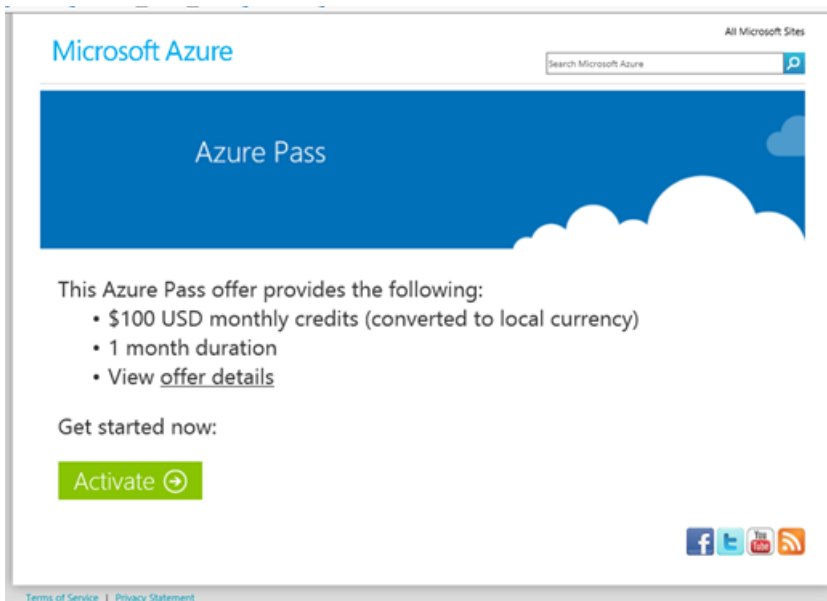
Phone*

Company*

☐ I would like to be contacted by Microsoft regarding Microsoft Azure developer offers and campaign.

Submit

4. Pulsa en Activate.



Microsoft Azure

Search Microsoft Azure

Azure Pass

This Azure Pass offer provides the following:

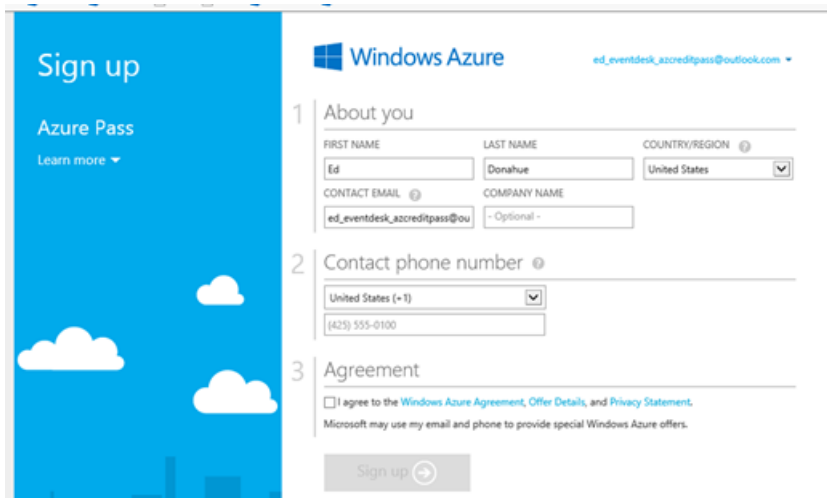
- \$100 USD monthly credits (converted to local currency)
- 1 month duration
- View [offer details](#)

Get started now:

Activate

Terms of Service | Privacy Statement

5. Completa el resto de información y pulsa en Sign Up.



Sign up

Azure Pass

Learn more

Windows Azure

ed_eventdesk_azcreditpass@outlook.com

1 About you

FIRST NAME Ed

LAST NAME Donahue

COUNTRY/REGION United States

CONTACT EMAIL ed_eventdesk_azcreditpass@ou

COMPANY NAME - Optional -

2 Contact phone number

United States (+1)

(425) 555-0100

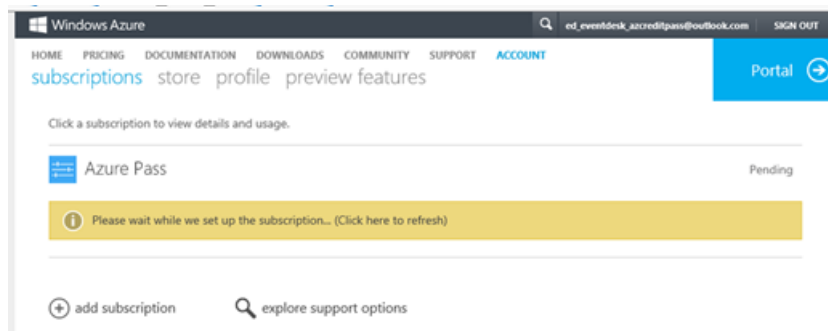
3 Agreement

☐ I agree to the [Windows Azure Agreement](#), [Offer Details](#), and [Privacy Statement](#).

Microsoft may use my email and phone to provide special Windows Azure offers.

Sign up

6. La suscripción tardará de 3 a 10 minutos en activarse.



7. Una vez activada te redireccionará a la página de resumen.

